

Reconfiguration of Cliques in a Graph

Takehiro Ito¹, Hirotaka Ono², and Yota Otachi³

¹ Graduate School of Information Sciences, Tohoku University,
Aoba-yama 6-6-05, Sendai, 980-8579, Japan.

takehiro@ecei.tohoku.ac.jp

² Faculty of Economics, Kyushu University,
Hakozaki 6-19-1, Higashi-ku, Fukuoka, 812-8581, Japan.

hirotaka@econ.kyushu-u.ac.jp

³ School of Information Science, JAIST,
Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan.

otachi@jaist.ac.jp

Abstract. We study reconfiguration problems for cliques in a graph, which determine whether there exists a sequence of cliques that transforms a given clique into another one in a step-by-step fashion. As one step of a transformation, we consider three different types of rules, which are defined and studied in reconfiguration problems for independent sets. We first prove that all the three rules are equivalent in cliques. We then show that the problems are PSPACE-complete for perfect graphs, while we give polynomial-time algorithms for several classes of graphs, such as even-hole-free graphs and cographs. In particular, the shortest variant, which computes the shortest length of a desired sequence, can be solved in polynomial time for chordal graphs, bipartite graphs, planar graphs, and bounded treewidth graphs.

1 Introduction

Recently, *reconfiguration problems* attract attention in the field of theoretical computer science. The problem arises when we wish to find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are also feasible and each step abides by a fixed reconfiguration rule (i.e., an adjacency relation defined on feasible solutions of the original problem). This kind of reconfiguration problem has been studied extensively for several well-known problems, including SATISFIABILITY [10], INDEPENDENT SET [3, 11, 12, 14, 22], VERTEX COVER [13, 16], CLIQUE, MATCHING [12], VERTEX-COLORING [2], and so on. (See also a recent survey [21].)

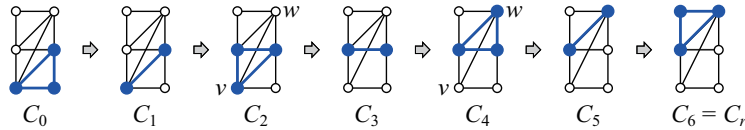


Fig. 1. A sequence $\langle C_0, C_1, \dots, C_6 \rangle$ of cliques in the same graph, where the vertices in cliques are depicted by large (blue) circles (tokens).

It is well known that independent sets, vertex covers and cliques are related with each other. Indeed, the well-known reductions for NP-completeness proofs are essentially the same for the three problems [7]. Despite reconfiguration problems for independent sets and vertex covers are two of the most well studied problems, we have only a few known results for reconfiguration problems for cliques (as we will explain later). In this paper, we thus investigate the complexity status of reconfiguration problems for cliques systematically, and show that the problems can be solved in polynomial time for a variety of graph classes, in contrast to independent sets and vertex covers.

1.1 Our problems and three rules

Recall that a *clique* of a graph $G = (V, E)$ is a vertex subset of G in which every two vertices are adjacent. (Figure 1 depicts seven different cliques in the same graph.) Suppose that we are given two cliques C_0 and C_r of G , and imagine that a token is placed on each vertex in C_0 . Then, we are asked to transform C_0 into C_r by abiding a prescribed reconfiguration rule on cliques. In this paper, we define three different reconfiguration rules on cliques, which were originally defined as the reconfiguration rules on independent sets [14], as follows:

- *Token Addition and Removal* (TAR rule): We can either add or remove a single token at a time if it results in a clique of size at least a given threshold $k \geq 0$. For example, in the sequence $\langle C_0, C_1, \dots, C_6 \rangle$ in Fig. 1, every two consecutive cliques follow the TAR rule for the threshold $k = 2$. In order to emphasize the threshold k , we sometimes call this rule the $\text{TAR}(k)$ rule.
- *Token Jumping* (TJ rule): A single token in a clique C can “jump” to any vertex in $V \setminus C$ if it results in a clique. For example, consider the sequence $\langle C_0, C_2, C_4, C_6 \rangle$ in Fig. 1, then two consecutive cliques C_{2i} and C_{2i+2} follow the TJ rule for each $i \in \{0, 1, 2\}$.
- *Token Sliding* (TS rule): We can slide a single token on a vertex v in a clique C to another vertex w in $V \setminus C$ if it results in a clique and there is an edge vw in G . For example, consider the sequence $\langle C_2, C_4 \rangle$ in Fig. 1, then two consecutive cliques C_2 and C_4 follow the TS rule, because v and w are adjacent.

A sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ of cliques of a graph G is called a *reconfiguration sequence* between two cliques C_0 and C_ℓ under $\text{TAR}(k)$ (or TJ, TS) if two consecutive cliques C_{i-1} and C_i follow the $\text{TAR}(k)$ (resp., TJ, TS) rule for all $i \in \{1, 2, \dots, \ell\}$. The *length* of a reconfiguration sequence is defined to be the number of cliques in the sequence minus one, that is, the length of $\langle C_0, C_1, \dots, C_\ell \rangle$ is ℓ .

Given two cliques C_0 and C_r of a graph G (and an integer $k \geq 0$ for TAR), CLIQUE RECONFIGURATION under TAR (or TJ, TS) is to determine whether there exists a reconfiguration sequence between C_0 and C_r under $\text{TAR}(k)$ (resp., TJ, TS). For example, consider the cliques C_0 and $C_r = C_6$ in Fig. 1; let $k = 2$ for TAR. Then, it is a yes-instance under the $\text{TAR}(2)$ and TJ rules as illustrated in Fig. 1, but is a no-instance under the TS rule.

In this paper, we also study the shortest variant, called **SHORTEST CLIQUE RECONFIGURATION**, under each of the three rules which computes the shortest length of a reconfiguration sequence between two given cliques under the rule. We define the shortest length to be infinity for a **no**-instance, and hence this variant is a generalization of **CLIQUE RECONFIGURATION**.

1.2 Known and related results

Ito et al. [12] introduced **CLIQUE RECONFIGURATION** under **TAR**, and proved that it is **PSPACE**-complete in general. They also considered the optimization problem of computing the maximum threshold k such that there is a reconfiguration sequence between two given cliques C_0 and C_r under **TAR**(k). This maximization problem cannot be approximated in polynomial time within any constant factor unless $P = NP$ [12].

INDEPENDENT SET RECONFIGURATION is one of the most well-studied reconfiguration problems, defined for independent sets in a graph. Kamiński et al. [14] studied the problem under **TAR**, **TJ** and **TS**. It is well known that a clique in a graph G forms an independent set in the complement \overline{G} of G , and vice versa. Indeed, some known results for **INDEPENDENT SET RECONFIGURATION** can be converted into ones for **CLIQUE RECONFIGURATION**. However, as far as we checked, only two results can be obtained for **CLIQUE RECONFIGURATION** by this conversion, because we take the complement of a graph. (These results will be formally discussed in Section 3.3.)

In this way, only a few results are known for **CLIQUE RECONFIGURATION**. In particular, there is almost no algorithmic result, and hence it is desired to develop efficient algorithms for the problem and its shortest variant.

1.3 Our contribution

In this paper, we embark on a systematic investigation of the computational status of **CLIQUE RECONFIGURATION** and its shortest variant. Figure 2 summarizes our results, which can be divided into the following four parts.

- (1) *Rule equivalence* (Section 3): We prove that all rules **TAR**, **TS** and **TJ** are equivalent in **CLIQUE RECONFIGURATION**. Then, any complexity result under one rule can be converted into the same complexity result under the other two rules. In addition, based on the rule equivalence, we show that **CLIQUE RECONFIGURATION** under any rule is **PSPACE**-complete for perfect graphs, and is solvable in linear time for cographs.
- (2) *Graphs with bounded clique size* (Section 4.1): We show that the shortest variant under any of **TAR**, **TS** and **TJ** can be solved in polynomial time for such graphs, which include bipartite graphs, planar graphs, and bounded treewidth graphs. Interestingly, **INDEPENDENT SET RECONFIGURATION** under any rule remains **PSPACE**-complete even for planar graphs [2, 11] and bounded treewidth graphs [22]. Therefore, this result shows a nice difference between the reconfiguration problems for cliques and independent sets.

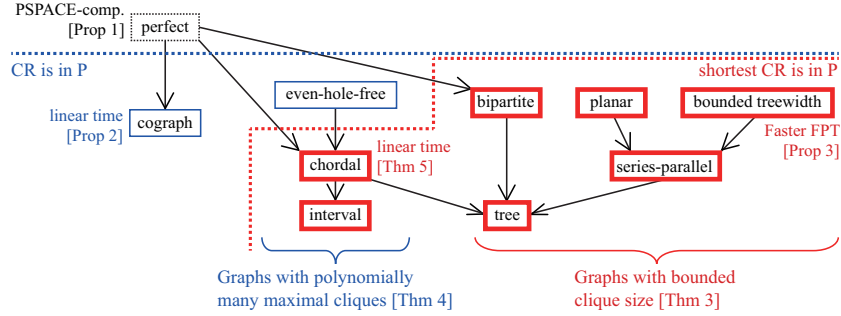


Fig. 2. Our results under all rules TAR, TS and TJ. Each arrow represents the inclusion relationship between graph classes: $A \rightarrow B$ represents that B is properly included in A [4]. Graph classes for which SHORTEST CLIQUE RECONFIGURATION is solvable in polynomial time are indicated by thick (red) boxes, while the ones for which CLIQUE RECONFIGURATION is solvable in polynomial time are indicated by thin (blue) boxes.

- (3) *Graphs with polynomially many maximal cliques* (Section 4.2): We show that CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved in polynomial time for such graphs, which include even-hole-free graphs, graphs of bounded boxicity, and K_t -subdivision-free graphs.
- (4) *Chordal graphs* (Section 5): We give a linear-time algorithm to solve the shortest variant under any of TAR, TS and TJ for chordal graphs. Note that the clique size of chordal graphs is not always bounded, and hence this result is independent from Result (2) above.

Several proofs move to appendices.

2 Preliminaries

In this section, we introduce some basic terms and notation.

2.1 Graph notation

In this paper, we assume without loss of generality that graphs are simple. For a graph G , we sometimes denote by $V(G)$ and $E(G)$ the vertex set and edge set of G , respectively. For a graph G , the *complement* \overline{G} of G is the graph such that $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{vw \mid v, w \in V(G), vw \notin E(G)\}$. We say that a graph class \mathcal{G} (i.e., a set of graphs) is *closed under taking complements* if $\overline{G} \in \mathcal{G}$ holds for every graph $G \in \mathcal{G}$.

In this paper, we deal with several graph classes systematically, and hence we do not define those graph classes precisely; we simply give the properties used for proving our results, with appropriate references.

2.2 Definitions for CLIQUE RECONFIGURATION

As explained in Introduction, we consider three (symmetric) adjacency relations on cliques in a graph. Let C_i and C_j be two cliques of a graph G . Then,

- $C_i \leftrightarrow C_j$ under TAR(k) for a nonnegative integer k if $|C_i| \geq k$, $|C_j| \geq k$, and $|C_i \Delta C_j| = |(C_i \setminus C_j) \cup (C_j \setminus C_i)| = 1$ hold;
- $C_i \leftrightarrow C_j$ under TJ if $|C_i| = |C_j|$, $|C_i \setminus C_j| = 1$, and $|C_j \setminus C_i| = 1$ hold; and
- $C_i \leftrightarrow C_j$ under TS if $|C_i| = |C_j|$, $C_i \setminus C_j = \{v\}$, $C_j \setminus C_i = \{w\}$, and $vw \in E(G)$ hold.

A sequence $\langle C_1, C_2, \dots, C_\ell \rangle$ of cliques of G is called a *reconfiguration sequence* between two cliques C_1 and C_ℓ under TAR(k) (or TJ, TS) if $C_{i-1} \leftrightarrow C_i$ holds under TAR(k) (resp., TJ, TS) for all $i \in \{2, 3, \dots, \ell\}$. A reconfiguration sequence under TAR(k) (or TJ, TS) is simply called a TAR(k)-sequence (resp., TJ-sequence, TS-sequence). We write $C_1 \rightsquigarrow C_\ell$ under TAR(k) (or TJ, TS) if there exists a TAR(k)-sequence (resp., TJ-sequence, TS-sequence) between C_1 and C_ℓ . Note that each clique in any TAR(k)-sequence is of size at least k , while all cliques in any TJ-sequence or TS-sequence have the same size. In addition, a reconfiguration sequence under any rule is *reversible*, that is, $C_1 \rightsquigarrow C_\ell$ if and only if $C_\ell \rightsquigarrow C_1$.

Let k be a nonnegative integer, and let C and C' be two cliques of a graph G . Then, we define $\text{TAR}(C, C', k)$, as follows:

$$\text{TAR}(C, C', k) = \begin{cases} \text{yes} & \text{if } C \rightsquigarrow C' \text{ under TAR}(k); \\ \text{no} & \text{otherwise.} \end{cases}$$

Given two cliques C_0 and C_r of a graph G and a nonnegative integer k , CLIQUE RECONFIGURATION under TAR is to compute $\text{TAR}(C_0, C_r, k)$. By the definition, $\text{TAR}(C_0, C_r, k) = \text{no}$ if $|C_0| < k$ or $|C_r| < k$ hold, and hence we may assume without loss of generality that both $|C_0| \geq k$ and $|C_r| \geq k$ hold; we call such an instance simply a TAR-instance, and denote it by (G, C_0, C_r, k) .

For two cliques C and C' of a graph G , we similarly define $\text{TJ}(C, C')$ and $\text{TS}(C, C')$. Given two cliques C_0 and C_r of G , we similarly define CLIQUE RECONFIGURATION under TJ and TS, and denote their instance by (G, C_0, C_r) . Then, we can assume that $|C_0| = |C_r|$ holds in a TJ- or a TS-instance (G, C_0, C_r) .

Given a TAR-instance (G, C_0, C_r, k) , let $\mathcal{C} = \langle C_0, C_1, \dots, C_\ell \rangle$ be a TAR(k)-sequence in G between C_0 and $C_r = C_\ell$. Then, the *length* of \mathcal{C} is defined to be the number of cliques in \mathcal{C} minus one, that is, the length of \mathcal{C} is ℓ . We denote by $\text{dist}_{\text{TAR}}(G, C_0, C_r, k)$ the minimum length of a TAR(k)-sequence in G between C_0 and C_r ; we let $\text{dist}_{\text{TAR}}(G, C_0, C_r, k) = +\infty$ if there is no TAR(k)-sequence in G between C_0 and C_r . The shortest variant, SHORTEST CLIQUE RECONFIGURATION, under TAR is to compute $\text{dist}_{\text{TAR}}(G, C_0, C_r, k)$. Similarly, we define $\text{dist}_{\text{TJ}}(G, C_0, C_r)$ and $\text{dist}_{\text{TS}}(G, C_0, C_r)$ for a TJ- and a TS-instance (G, C_0, C_r) , respectively. Then, SHORTEST CLIQUE RECONFIGURATION under TJ or TS is defined similarly. We sometimes drop G and simply write $\text{dist}_{\text{TAR}}(C_0, C_r, k)$, $\text{dist}_{\text{TJ}}(C_0, C_r)$ and $\text{dist}_{\text{TS}}(C_0, C_r)$ if it is clear from context.

We note that CLIQUE RECONFIGURATION under any rule is a decision problem asking for the existence of a reconfiguration sequence, and its shortest variant asks for simply computing the shortest length of a reconfiguration sequence. Therefore, the problems do not ask for an actual reconfiguration sequence. How-

ever, our algorithms proposed in this paper can be easily modified so that they indeed find a reconfiguration sequence.

3 Rule Equivalence and Complexity

In this section, we first prove that all three rules TAR, TS and TJ are equivalent in CLIQUE RECONFIGURATION. We then discuss some complexity results that can be obtained from known results for INDEPENDENT SET RECONFIGURATION.

3.1 Equivalence of TS and TAR rules

TS and TAR rules are equivalent, as in the following sense.

Theorem 1. *TS and TAR rules are equivalent in CLIQUE RECONFIGURATION, as follows:*

- (a) *for any TS-instance (G, C_0, C_r) , a TAR-instance (G, C'_0, C'_r, k') can be constructed in linear time such that $\text{TS}(C_0, C_r) = \text{TAR}(C'_0, C'_r, k')$ and $\text{dist}_{\text{TS}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C'_0, C'_r, k')/2$; and*
- (b) *for any TAR-instance (G, C_0, C_r, k) , a TS-instance (G, C'_0, C'_r) can be constructed in linear time such that $\text{TAR}(C_0, C_r, k) = \text{TS}(C'_0, C'_r)$.*

By Theorem 1(a), note that the reduction from TS to TAR preserves the shortest length of reconfiguration sequences.

Proof of Theorem 1(a). Let (G, C_0, C_r) be a TS-instance with $|C_0| = |C_r| = k$. Then, as the corresponding TAR-instance (G, C'_0, C'_r, k') , we let $C'_0 = C_0$, $C'_r = C_r$ and $k' = k$; this TAR-instance can be clearly constructed in linear time. We thus prove the following lemma, as a proof of Theorem 1(a).

Lemma 1. *Let G be a graph, and let C_0 and C_r be any pair of cliques of G such that $|C_0| = |C_r| = k$. Then, $\text{TS}(C_0, C_r) = \text{TAR}(C_0, C_r, k)$ and $\text{dist}_{\text{TS}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C_0, C_r, k)/2$.*

Proof of Theorem 1(b). Let (G, C_0, C_r, k) be a TAR-instance; note that $|C_0| \neq |C_r|$ may hold, and both $|C_0| \geq k$ and $|C_r| \geq k$ hold. Then, as the corresponding TS-instance (G, C'_0, C'_r) , let $C'_0 \subseteq C_0$ and $C'_r \subseteq C_r$ be arbitrary subsets of size exactly k ; this TS-instance can be clearly constructed in linear time. We thus prove the following lemma, as a proof of Theorem 1(b).

Lemma 2. *Let (G, C_0, C_r, k) be a TAR-instance, and let $C'_0 \subseteq C_0$ and $C'_r \subseteq C_r$ be arbitrary subsets of size exactly k . Then, $\text{TAR}(C_0, C_r, k) = \text{TS}(C'_0, C'_r)$.*

3.2 Equivalence of TJ and TAR rules

TJ and TAR rules are equivalent, as in the following sense.

Theorem 2. *TJ and TAR rules are equivalent in CLIQUE RECONFIGURATION, as follows:*

- (a) for any TJ-instance (G, C_0, C_r) , a TAR-instance (G, C'_0, C'_r, k') can be constructed in linear time such that $\text{TJ}(C_0, C_r) = \text{TAR}(C'_0, C'_r, k')$ and $\text{dist}_{\text{TJ}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C'_0, C'_r, k')/2$; and
- (b) for any TAR-instance (G, C_0, C_r, k) , a TJ-instance (G, C'_0, C'_r) can be constructed in linear time such that $\text{TAR}(C_0, C_r, k) = \text{TJ}(C'_0, C'_r)$.

By Theorem 2(a), note that the reduction from TJ to TAR preserves the shortest length of reconfiguration sequences.

Proof of Theorem 2(a). Let (G, C_0, C_r) be a TJ-instance with $|C_0| = |C_r| = k$. Then, as the corresponding TAR-instance (G, C'_0, C'_r, k') , we let $C'_0 = C_0$, $C'_r = C_r$ and $k' = k - 1$; this TAR-instance can be clearly constructed in linear time. We thus prove the following lemma, as a proof of Theorem 2(a).

Lemma 3. *Let G be a graph, and let C_0 and C_r be any pair of cliques of G such that $|C_0| = |C_r| = k$. Then, $\text{TJ}(C_0, C_r) = \text{TAR}(C_0, C_r, k - 1)$ and $\text{dist}_{\text{TJ}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C_0, C_r, k - 1)/2$.*

Proof of Theorem 2(b). Let (G, C_0, C_r, k) be a TAR-instance; $|C_0| \neq |C_r|$ may hold, and both $|C_0| \geq k$ and $|C_r| \geq k$ hold. We first give the following lemma.

Lemma 4. *Let (G, C_0, C_r, k) be a TAR-instance such that $C_0 \neq C_r$. Suppose that there exists an index $j \in \{0, r\}$ such that $|C_j| = k$ and C_j is a maximal clique in G . Then, $\text{TAR}(C_0, C_r, k) = \text{no}$.*

Proof. Since C_j is maximal, there is no clique in G which can be obtained by adding a vertex to C_j . Furthermore, since $|C_j| = k$, we cannot delete any vertex from C_j to keep the threshold k . Thus, there is no clique C in G such that $C_j \leftrightarrow C$ under $\text{TAR}(k)$. Since $C_0 \neq C_r$, we have $\text{TAR}(C_0, C_r, k) = \text{no}$. \square

We thus assume without loss of generality that none of C_0 and C_r is a maximal clique in G of size k ; note that the maximality of a clique can be determined in linear time. Then, we construct the corresponding TJ-instance (G, C'_0, C'_r) , as in the following two cases (i) and (ii):

- (i) for each $j \in \{0, r\}$ such that $|C_j| \geq k + 1$, let $C'_j \subseteq C_j$ be an arbitrary subset of size exactly $k + 1$; and
- (ii) for each $j \in \{0, r\}$ such that $|C_j| = k$, let $C'_j \supset C_j$ be an arbitrary superset of size exactly $k + 1$.

This TJ-instance can be clearly constructed in linear time. We thus prove the following lemma, as a proof of Theorem 2(b).

Lemma 5. *Let (G, C_0, C_r, k) be a TAR-instance, and let (G, C'_0, C'_r) be the corresponding TJ-instance constructed above. Then, $\text{TAR}(C_0, C_r, k) = \text{TJ}(C'_0, C'_r)$.*

3.3 Results obtained from INDEPENDENT SET RECONFIGURATION

We here show two complexity results for CLIQUE RECONFIGURATION, which can be obtained from known results for INDEPENDENT SET RECONFIGURATION.

Consider a vertex subset C of a graph G . Then, C forms a clique in G if and only if C forms an independent set in the complement \overline{G} of G . Therefore, the following lemma clearly holds.

Lemma 6. *Let G be a graph, and let C_j be a clique of G for each $j \in \{0, 1, \dots, \ell\}$. Then, $\langle C_0, C_1, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence of cliques in G if and only if $\langle C_0, C_1, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence of independent sets in the complement \overline{G} of G .*

By Lemma 6 we can convert a complexity result for INDEPENDENT SET RECONFIGURATION under TAR for a graph class \mathcal{G} into one for CLIQUE RECONFIGURATION under TAR for \mathcal{G} if the graph class \mathcal{G} is closed under taking complements. Note that, by Theorems 1 and 2, any complexity result under one rule can be converted into the same complexity result under the other two rules.

Proposition 1. *CLIQUE RECONFIGURATION is PSPACE-complete for perfect graphs under all rules TAR, TS and TJ.*

Proposition 2. *CLIQUE RECONFIGURATION can be solved in linear time for cographs under all rules TAR, TS and TJ.*

4 Polynomial-Time Algorithms

In this section, we show that CLIQUE RECONFIGURATION is solvable in polynomial time for several graph classes. We deal with two types of graph classes, that is, graphs of bounded clique size (in Section 4.1) and graphs having polynomially many maximal cliques (in Section 4.2).

4.1 Graphs of bounded clique size

In this subsection, we show that SHORTEST CLIQUE RECONFIGURATION can be solved in polynomial time for graphs of bounded clique size; as we will explain later, such graphs include bipartite graphs, planar graphs, and graphs of bounded treewidth. For a graph G , we denote by $\omega(G)$ the size of a maximum clique in G . Then, we have the following theorem.

Theorem 3. *Let G be a graph with n vertices such that $\omega(G) \leq w$ for a positive integer w . Then, SHORTEST CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved in time $O(w^2 n^w)$ for G .*

It is well known that $\omega(G) \leq 4$ for any planar graph G , and $\omega(G') \leq 2$ for any bipartite graph G' . We thus have the following corollary.

Corollary 1. *SHORTEST CLIQUE RECONFIGURATION under TAR, TS and TJ can be solved in polynomial time for planar graphs and bipartite graphs.*

By the definition of treewidth [1], we have $\omega(G) \leq t+1$ for any graph G whose treewidth can be bounded by a positive integer t . By Theorem 3 this observation gives an $O(t^2 n^{t+1})$ -time algorithm for SHORTEST CLIQUE RECONFIGURATION. However, for this case, we can obtain a faster fixed-parameter algorithm, where the parameter is the treewidth t , as follows.

Proposition 3. *Let G be a graph with n vertices whose treewidth is bounded by a positive integer t . Then, SHORTEST CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved for G in time $O(c^t n)$, where c is some constant.*

Proposition 3 implies that SHORTEST CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved in time $O(c^w n)$ for chordal graphs G when parameterized by the size w of a maximum clique in G , where n is the number of vertices in G and c is some constant; because the treewidth of a chordal graph G can be bounded by the size of a maximum clique in G minus one [17]. However, we give a linear-time algorithm to solve the shortest variant under any rule for chordal graphs in Section 5.

4.2 Graphs with polynomially many maximal cliques

In this subsection, we consider the class of graphs having polynomially many maximal cliques, which properly contains the class of graphs with bounded clique size (in Section 4.1). Note that, even if a graph G has a polynomial number of maximal cliques, G may have a super-polynomial number of cliques.

Theorem 4. *Let G be a graph with n vertices and m edges, and let $\mathcal{M}(G)$ be the set of all maximal cliques in G . Then, CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved for G in time $O(mn|\mathcal{M}(G)| + n|\mathcal{M}(G)|^2)$.*

Before proving Theorem 4, we give the following corollary.

Corollary 2. *CLIQUE RECONFIGURATION under TAR, TS and TJ can be solved in polynomial time for even-hole-free graphs, graphs of bounded boxicity, and K_t -subdivision-free graphs.*

Proof. By Theorem 4 it suffices to show that the claimed graphs have polynomially many maximal cliques. Polynomial bounds on the number of maximal cliques are shown for even-hole-free graphs in [5], for graphs of bounded boxicity in [18], and for K_t -subdivision-free graphs in [15]. \square

In this subsection, we prove Theorem 4. However, by Theorems 1(a) and 2(a) it suffices to give such an algorithm only for the TAR rule.

Let (G, C_0, C_r, k) be any TAR-instance. Then, we define the k -intersection maximal-clique graph of G , denoted by $\text{MC}_k(G)$, as follows:

- (i) each node in $\text{MC}_k(G)$ corresponds to a clique in $\mathcal{M}(G)$; and
- (ii) two nodes in $\text{MC}_k(G)$ are joined by an edge if and only if $|M \cap M'| \geq k$ holds for the corresponding two maximal cliques M and M' in $\mathcal{M}(G)$.

Note that any maximal clique in $\mathcal{M}(G)$ of size less than k is contained in $\text{MC}_k(G)$ as an isolated node. We now give the key lemma to prove Theorem 4.

Lemma 7. *Let G be a graph, and let C and C' be any pair of cliques in G such that $|C| \geq k$ and $|C'| \geq k$. Let $M \supseteq C$ and $M' \supseteq C'$ be arbitrary maximal cliques in $\mathcal{M}(G)$. Then, $C \rightsquigarrow C'$ under $\text{TAR}(k)$ if and only if $\text{MC}_k(G)$ contains a path between the two nodes corresponding to M and M' .*

Proof of Theorem 4.

For any graph G with n vertices and m edges, Tsukiyama et al. [19] proved that the set $\mathcal{M}(G)$ can be computed in time $O(mn|\mathcal{M}(G)|)$. Thus, we can construct $\text{MC}_k(G)$ in time $O(mn|\mathcal{M}(G)| + n|\mathcal{M}(G)|^2)$. By the breadth-first search on $\text{MC}_k(G)$ which starts from an arbitrary maximal clique (node) $M \supseteq C_0$, we can check in time $O(|\mathcal{M}(G)|^2)$ whether $\text{MC}_k(G)$ has a path to a maximal clique $M' \supseteq C_r$. Then, the theorem follows from Lemma 7. \square

5 Linear-Time Algorithm for Chordal Graphs

Since any chordal graph is even-hole free, by Corollary 2 `CLIQUE RECONFIGURATION` is solvable in polynomial time for chordal graphs. Furthermore, we have discussed in Section 4.1 that the shortest variant is fixed-parameter tractable for chordal graphs when parameterized by the size of a maximum clique in a graph. However, we give the following theorem in this section.

Theorem 5. *SHORTEST CLIQUE RECONFIGURATION under any of TAR, TS and TJ can be solved in linear time for chordal graphs.*

In this section, we prove Theorem 5. By Theorems 1(a) and 2(a) it suffices to give a linear-time algorithm for a TAR-instance; recall that the reduction from TS/TJ to TAR preserves the shortest length of reconfiguration sequences.

Our algorithm consists of two phases. The first is a linear-time reduction from a given TAR-instance (G, C_0, C_r, k) for a chordal graph G to a TAR-instance (H, C_0, C_r, k) for an interval graph H such that $\text{dist}_{\text{TAR}}(H, C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$. The second is a linear-time algorithm for interval graphs.

Definitions of chordal graphs and interval graphs.

A graph is a *chordal graph* if every induced cycle is of length three. Recall that $\mathcal{M}(G)$ is the set of all maximal cliques in a graph G , and we denote by $\mathcal{M}(G; v)$ the set of all maximal cliques in G that contain a vertex $v \in V(G)$. A tree \mathcal{T} is a *clique tree* of a graph G if it satisfies the following conditions:

- each node in \mathcal{T} corresponds to a maximal clique in $\mathcal{M}(G)$; and
- for each $v \in V(G)$, the subgraph of \mathcal{T} induced by $\mathcal{M}(G; v)$ is connected.

It is known that a graph is a chordal graph if and only if it has a clique tree [8]. A clique tree of a chordal graph can be computed in linear time (see [18, §15.1]).

A graph is an *interval graph* if it can be represented as the intersection graph of intervals on the real line. A *clique path* is a clique tree which is a path. It is known that a graph is an interval graph if and only if it has a clique path [6, 9].

5.1 Linear-time reduction from chordal graphs to interval graphs

In this subsection, we describe the first phase of our algorithm.

Let (G, C_0, C_r, k) be any TAR-instance for a chordal graph G , and let \mathcal{T} be a clique tree of G . Then, we find an arbitrary pair of maximal cliques M_0 and M_t in G (i.e., two nodes in \mathcal{T}) such that $C_0 \subseteq M_0$ and $C_r \subseteq M_t$. Let (M_0, M_1, \dots, M_t)

be the unique path in \mathcal{T} from M_0 to M_t . We define a graph H' as the subgraph of G induced by the maximal cliques M_0, M_1, \dots, M_t . Note that H' is an interval graph, because (M_0, M_1, \dots, M_t) forms a clique path.

The following lemma implies that the interval graph H' has a $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_{\ell'} \rangle$ such that $\ell' = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$, and hence yields that $\text{dist}_{\text{TAR}}(H', C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$ holds.

Lemma 8. *Let (G, C_0, C_r, k) be a TAR -instance for a chordal graph G , and let \mathcal{T} be a clique tree of G . Suppose that $\langle C_0, C_1, \dots, C_{\ell} \rangle$ is a shortest $\text{TAR}(k)$ -sequence in G from C_0 to $C_{\ell} = C_r$. Let (M_0, M_1, \dots, M_t) be the path in \mathcal{T} from M_0 to M_t for any pair of maximal cliques $M_0 \supseteq C_0$ and $M_t \supseteq C_r$. Then, there is a monotonically increasing function $f: \{0, 1, \dots, \ell\} \rightarrow \{0, 1, \dots, t\}$ such that $C_i \subseteq M_{f(i)}$ for each $i \in \{0, 1, \dots, \ell\}$.*

Although Lemma 8 implies that $\text{dist}_{\text{TAR}}(H', C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$ holds for the interval graph H' , it seems difficult to find two maximal cliques $M_0 \supseteq C_0$ and $M_t \supseteq C_r$ (and hence construct H' from G) in linear time. However, by a small trick, we can construct an interval graph H in linear time such that $\text{dist}_{\text{TAR}}(H, C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$, as follows.

Lemma 9. *Given a TAR -instance (G, C_0, C_r, k) for a chordal graph G , one can obtain a subgraph H of G in linear time such that H is an interval graph, $C_0, C_r \subseteq V(H)$ and $\text{dist}_{\text{TAR}}(H, C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$.*

5.2 Linear-time algorithm for interval graphs

In this subsection, we describe the second phase of our algorithm.

Let H be a given interval graph, and we assume that its clique path \mathcal{P} has $V(\mathcal{P}) = \mathcal{M}(H) = \{M_0, M_1, \dots, M_t\}$ and $E(\mathcal{P}) = \{\{M_i, M_{i+1}\} \mid 0 \leq i < t\}$. Note that we can assume that $t \geq 1$, that is, H has at least two maximal cliques; otherwise we can easily solve the problem in linear time (as in Lemma 12 in Appendix C.1). For a vertex v in H , let $l_v = \min\{i \mid v \in M_i\}$ and $r_v = \max\{i \mid v \in M_i\}$; the indices l_v and r_v are called the *l-value* and *r-value* of v , respectively. Note that $v \in M_i$ if and only if $l_v \leq i \leq r_v$. For an interval graph H , such a clique path \mathcal{P} and the indices l_v and r_v for all vertices $v \in V(H)$ can be computed in linear time [20].

Let (H, C_0, C_r, k) be a TAR -instance. We assume that $C_0 \subseteq M_0$, $C_0 \not\subseteq M_1$ and $C_r \subseteq M_t$; otherwise, we can remove the maximal cliques M_i with $i < \min\{r_v \mid v \in C_0\}$ and $i > \max\{l_v \mid v \in C_r\}$ in linear time. Our algorithm greedily constructs a shortest $\text{TAR}(k)$ -sequence from C_0 to C_r , as follows:

- (1) if $C_0 \not\subseteq C_r$ and $|C_0| \geq k + 1$, then remove a vertex with the minimum r -value in $C_0 \setminus C_r$ from C_0 ;
- (2) otherwise add a vertex in $(C_r \setminus C_0) \cap M_0$ if any; if no such vertex exists, add a vertex with the maximum r -value in $M_0 \setminus C_0$.

We regard the clique obtained by the operations above as C_0 ; if necessary, we shift the indices of M_i so that $C_0 \subseteq M_0$ and $C_0 \not\subseteq M_1$ hold; and repeat. If $C_0 \neq C_r$ and none of the operations above is possible, we can conclude that

(H, C_0, C_r, k) is a no-instance. The correctness proof of this greedy algorithm and the estimation of its running time can be found in Appendix C.3.

This completes the proof of Theorem 5.

6 Conclusion

In this paper, we have systematically shown that CLIQUE RECONFIGURATION and its shortest variant can be solved in polynomial time for several graph classes. As far as we know, this is the first example of a reconfiguration problem such that it is PSPACE-complete in general, but is solvable in polynomial time for such a variety of graph classes.

Acknowledgments

This work is partially supported by MEXT/JSPS KAKENHI 25106504 and 25330003 (T. Ito), 25104521, 26540005 and 26540005 (H. Ono), and 24106004 and 25730003 (Y. Otachi).

References

1. Bodlaender, H.L., Drange, P.G., Dregi, M.S., Fomin, F.V., Lokshtanov, D., Pilipczuk, M.: An $O(c^k n)$ 5-approximation algorithm for treewidth. Proc. of FOCS 2013, pp. 499–508 (2013)
2. Bonsma, P., Cereceda, L.: Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. Theoretical Computer Science 410, pp. 5215–5226 (2009)
3. Bonsma, P.: Independent set reconfiguration in cographs. Proc. of WG 2014, LNCS 8747, pp. 105–116 (2014)
4. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey, SIAM (1999)
5. da Silva, M.V.G., Vušković, K.: Triangulated neighborhoods in even-hole-free graphs. *Discrete Mathematics*, 307:1065–1073, 2007.
6. Fulkerson, D.R., Gross, O.A.: Incidence matrices and interval graphs. Pacific J. Mathematics 15, pp. 835–855 (1965)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
8. Gavril, F.: The intersection graphs of subtrees in trees are exactly the chordal graphs. J. Combinatorial Theory, Series B 16, pp. 47–56 (1974)
9. Gilmore, P.C., Hoffman, A.J.: A characterization of comparability graphs and of interval graphs. Canadian J. Mathematics 16, pp. 539–548 (1964)
10. Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of Boolean satisfiability: computational and structural dichotomies. SIAM J. Computing 38, pp. 2330–2355 (2009)
11. Hearn, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. Theoretical Computer Science 343, pp. 72–96 (2005)
12. Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theoretical Computer Science 412, pp. 1054–1065 (2011)

13. Ito, T., Nooka, H., Zhou, X.: Reconfiguration of vertex covers in a graph. To appear in Proc. of IWOCA 2014.
14. Kamiński, M., Medvedev, P., Milanič, M.: Complexity of independent set reconfigurability problems. *Theoretical Computer Science* 439, pp. 9–15 (2012)
15. Lee, C., Oum, S.: Number of cliques in graphs with forbidden subdivision. [arXiv:1407.7707](#) (2014)
16. Mouawad, A.E., Nishimura, N., Raman, V.: Vertex cover reconfiguration and beyond. Proc. of ISAAC 2014, LNCS 8889, pp. 452–463 (2014)
17. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms* 7, pp. 309–322 (1986)
18. Spinrad, J.P.: *Efficient Graph Representations*. American Mathematical Society (2003)
19. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. *SIAM J. Computing* 6, pp. 505–517 (1977)
20. Uehara, R., Uno, Y.: On computing longest paths in small graph classes. *International J. Foundations of Computer Science* 18, pp. 911–930 (2007)
21. van den Heuvel, J.: The complexity of change. *Surveys in Combinatorics 2013*, London Mathematical Society Lecture Notes Series 409 (2013).
22. Wrochna, M.: Reconfiguration in bounded bandwidth and treedepth. [arXiv:1405.0847](#) (2014)

A Proofs Omitted from Section 3

A.1 Proof of Lemma 1

To prove Lemma 1, we first give the following lemma.

Lemma 10. *Let G be a graph, and let C and C' be any pair of cliques of G such that $|C| = |C'| = k$ and $C \rightsquigarrow C'$ under $\text{TAR}(k)$. Then, there exists a shortest $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ from $C_0 = C$ to $C_\ell = C'$ such that $|C_{2i-1}| = k + 1$ and $|C_{2i}| = k$ for every $i \in \{1, 2, \dots, \ell/2\}$.*

Proof. Let $\langle C_0, C_1, \dots, C_\ell \rangle$ be a shortest $\text{TAR}(k)$ -sequence from $C_0 = C$ to $C_\ell = C'$ which minimizes the sum $\sum_{i=0}^{\ell} |C_i|$. Since each clique in the $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ is of size at least k , it suffices to show that $|C_j| \leq k + 1$ holds for every $j \in \{1, 2, \dots, \ell - 1\}$.

Let s be an index satisfying $|C_s| = \max_{i=0}^{\ell} |C_i|$, and suppose for a contradiction that $|C_s| \geq k + 2$. By the definition of s , we have $C_{s-1} \subset C_s \supset C_{s+1}$. Let $C_s = C_{s-1} \cup \{a\}$ and $C_{s+1} = (C_{s-1} \cup \{a\}) \setminus \{b\}$. Note that, since $\langle C_0, C_1, \dots, C_\ell \rangle$ is shortest, we have $a \neq b$ and hence $b \in C_{s-1}$. We now replace the clique C_s by another clique $C'_s = C_{s-1} \setminus \{b\}$, and obtain the following sequence C' of cliques:

$$C' = \langle C_0, C_1, \dots, C_{s-1}, C_{s-1} \setminus \{b\}, C_{s+1}, \dots, C_\ell \rangle.$$

Since $C_{s-1} = C_s \setminus \{a\}$ and $|C_s| \geq k + 2$, we have $|C'_s| = |C_s \setminus \{a, b\}| \geq k$ and hence $C_{s-1} \leftrightarrow C_{s+1} \setminus \{b\} = C'_s$ under $\text{TAR}(k)$. Furthermore, since $C_{s+1} = (C_{s-1} \cup \{a\}) \setminus \{b\} = C'_s \cup \{a\}$, we have $C'_s \leftrightarrow C_{s+1}$ under $\text{TAR}(k)$. Therefore, C' is a $\text{TAR}(k)$ -sequence between C and C' .

Note that C' is of length ℓ , and hence it is a shortest $\text{TAR}(k)$ -sequence between C and C' . Since $C'_s = C_s \setminus \{a, b\}$, we have $|C'_s| < |C_s|$ and hence

$$|C'_s| + \sum \left\{ |C_j| : j \in \{0, 1, \dots, \ell\} \setminus \{s\} \right\} < \sum_{i=0}^{\ell} |C_i|.$$

This contradicts the assumption that $\langle C_0, C_1, \dots, C_\ell \rangle$ is a shortest $\text{TAR}(k)$ -sequence from $C_0 = C$ to $C_\ell = C'$ which minimizes the sum $\sum_{i=0}^{\ell} |C_i|$. \square

Proof of Lemma 1.

We first prove that $\text{TAR}(C_0, C_r, k) = \text{yes}$ if $\text{TS}(C_0, C_r) = \text{yes}$. In this case, there exists a TS -sequence between C_0 and C_r ; let $\langle C_0, C_1, \dots, C_\ell \rangle$ be a shortest one, that is, $C_\ell = C_r$ and $\ell = \text{dist}_{\text{TS}}(C_0, C_r)$. Then, since this is a TS -sequence, we have $u_{j-1}w_j \in E(G)$ for each $j \in \{1, 2, \dots, \ell\}$, where $C_{j-1} \setminus C_j = \{u_{j-1}\}$ and $C_j \setminus C_{j-1} = \{w_j\}$. (See Fig. 3(a).) Therefore, $C_{j-1} \cup C_j (= C_{j-1} \cup \{w_j\})$ forms a clique of size $k + 1$. Then, for each $j \in \{1, 2, \dots, \ell\}$, we replace each sub-sequence $\langle C_j \rangle$ with $\langle C_{j-1} \cup \{w_j\}, C_j \rangle$, and obtain the following sequence C' of cliques:

$$C' = \langle C_0, C_0 \cup \{w_1\}, C_1, \dots, C_{j-1} \cup \{w_j\}, C_j, \dots, C_{\ell-1} \cup \{w_\ell\}, C_\ell \rangle.$$



Fig. 3. Illustration for Lemma 1.

Notice that $C_{j-1} \cup \{w_j\} \leftrightarrow C_j$ under $\text{TAR}(k)$ for each $j \in \{1, 2, \dots, \ell\}$, because $(C_{j-1} \cup \{w_j\}) \setminus \{u_{j-1}\} = C_j$. Therefore, the sequence \mathcal{C}' above is a $\text{TAR}(k)$ -sequence from C_0 to $C_\ell = C_r$, and hence $\text{TAR}(C_0, C_r, k) = \text{yes}$. Furthermore, by the construction, \mathcal{C}' is of length 2ℓ . Therefore, we have

$$\text{dist}_{\text{TAR}}(C_0, C_r, k) \leq 2\ell = 2 \cdot \text{dist}_{\text{TS}}(C_0, C_r). \quad (1)$$

We then prove that $\text{TS}(C_0, C_r) = \text{yes}$ if $\text{TAR}(C_0, C_r, k) = \text{yes}$. In this case, there exists a $\text{TAR}(k)$ -sequence between C_0 and C_r ; let $\langle C_0, C_1, \dots, C_{\ell'} \rangle$ be a shortest one, that is, $C_{\ell'} = C_r$ and $\ell' = \text{dist}_{\text{TAR}}(C_0, C_r, k)$. Furthermore, by Lemma 10 we can assume that $|C_{2i-1}| = k+1$ and $|C_{2i}| = k$ for every $i \in \{1, 2, \dots, \ell'/2\}$. Then, observe that $C_{2i-1} = C_{2i-2} \cup C_{2i}$ for every $i \in \{1, 2, \dots, \ell'/2\}$, and let $C_{2i-1} = C_{2i-2} \cup \{w_{2i-1}\} = C_{2i} \cup \{u_{2i-1}\}$. (See Fig. 3(b).) Since this $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_{\ell'} \rangle$ is shortest, we have $u_{2i-1} \neq w_{2i-1}$. Furthermore, since both u_{2i-1} and w_{2i-1} belong to the clique C_{2i-1} , they are adjacent. Therefore, for every $i \in \{1, 2, \dots, \ell'/2\}$, we have $C_{2i-2} \leftrightarrow C_{2i}$ under TS ; we replace each sub-sequence $\langle C_{2i-1}, C_{2i} \rangle$ with $\langle C_{2i} \rangle$, and obtain $\mathcal{C}'' = \langle C_0, C_2, C_4, \dots, C_{\ell'} \rangle$. In this way, \mathcal{C}'' is a TS -sequence from C_0 to $C_{\ell'} = C_r$, and hence $\text{TS}(C_0, C_r) = \text{yes}$. Furthermore, the length of \mathcal{C}'' is $\ell'/2$, and hence

$$\text{dist}_{\text{TS}}(C_0, C_r) \leq \ell'/2 = \text{dist}_{\text{TAR}}(C_0, C_r, k)/2. \quad (2)$$

By Eqs. (1) and (2) we have $\text{dist}_{\text{TS}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C_0, C_r, k)/2$. \square

A.2 Proof of Lemma 2

Since $C'_0 \subseteq C_0$ and $|C'_0| = k$, we have $C_0 \rightsquigarrow C'_0$ under $\text{TAR}(k)$ by deleting the vertices in $C_0 \setminus C'_0$ from C_0 one by one. Similarly, we have $C'_r \rightsquigarrow C_r$ under $\text{TAR}(k)$; recall that any reconfiguration sequence is reversible. Since $|C'_0| = |C'_r| = k$, by Lemma 1 we have

$$\text{TS}(C'_0, C'_r) = \text{TAR}(C'_0, C'_r, k). \quad (3)$$

We now prove that $\text{TAR}(C_0, C_r, k) = \text{yes}$ if $\text{TS}(C'_0, C'_r) = \text{yes}$. In this case, by Eq. (3) we have $\text{TAR}(C'_0, C'_r, k) = \text{yes}$ and hence $C'_0 \rightsquigarrow C'_r$ under $\text{TAR}(k)$. Thus, $C_0 \rightsquigarrow C'_0 \rightsquigarrow C'_r \rightsquigarrow C_r$ holds under $\text{TAR}(k)$, and hence $\text{TAR}(C_0, C_r, k) = \text{yes}$.

We finally prove that $\text{TS}(C'_0, C'_r) = \text{yes}$ if $\text{TAR}(C_0, C_r, k) = \text{yes}$. In this case, since $\text{TAR}(C_0, C_r, k) = \text{yes}$, we have $C_0 \rightsquigarrow C_r$ under $\text{TAR}(k)$. Therefore, $C'_0 \rightsquigarrow C_0 \rightsquigarrow C_r \rightsquigarrow C'_r$ holds under $\text{TAR}(k)$, and hence $\text{TAR}(C'_0, C'_r, k) = \text{yes}$. By Eq. (3) we then have $\text{TS}(C'_0, C'_r) = \text{yes}$. \square

A.3 Proof of Lemma 3

We first give the following lemma, which can be obtained from the same arguments as in Lemma 10 by just shifting the threshold by one.

Lemma 11. *Let G be a graph, and let C and C' be any pair of cliques of G such that $|C| = |C'| = k$ and $C \rightsquigarrow C'$ under $\text{TAR}(k-1)$. Then, there exists a shortest $\text{TAR}(k-1)$ -sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ from $C_0 = C$ to $C_\ell = C'$ such that $|C_{2i-1}| = k-1$ and $|C_{2i}| = k$ for every $i \in \{1, 2, \dots, \ell/2\}$.*

Proof of Lemma 3.

We first prove that $\text{TAR}(C_0, C_r, k-1) = \text{yes}$ if $\text{TJ}(C_0, C_r) = \text{yes}$. In this case, there exists a TJ -sequence between C_0 and C_r ; let $\langle C_0, C_1, \dots, C_\ell \rangle$ be a shortest one, that is, $C_\ell = C_r$ and $\ell = \text{dist}_{\text{TJ}}(C_0, C_r)$. For each $j \in \{1, 2, \dots, \ell\}$, let $C_{j-1} \setminus C_j = \{u_{j-1}\}$ and $C_j \setminus C_{j-1} = \{w_j\}$. Then, we replace each sub-sequence $\langle C_j \rangle$ with $\langle C_{j-1} \setminus \{u_{j-1}\}, C_j \rangle$ for each $j \in \{1, 2, \dots, \ell\}$, and obtain the following sequence \mathcal{C}' of cliques:

$$\mathcal{C}' = \langle C_0, C_0 \setminus \{u_0\}, C_1, \dots, C_{j-1} \setminus \{u_{j-1}\}, C_j, \dots, C_{\ell-1} \setminus \{u_{\ell-1}\}, C_\ell \rangle.$$

Notice that $C_{j-1} \setminus \{u_{j-1}\} \leftrightarrow C_j$ under $\text{TAR}(k-1)$ for each $j \in \{1, 2, \dots, \ell\}$, because $(C_{j-1} \setminus \{u_{j-1}\}) \cup \{w_j\} = C_j$ and $|C_{j-1} \setminus \{u_{j-1}\}| = k-1$. Therefore, the sequence \mathcal{C}' above is a $\text{TAR}(k-1)$ -sequence from C_0 to $C_\ell = C_r$, and hence $\text{TAR}(C_0, C_r, k-1) = \text{yes}$. Furthermore, by the construction, \mathcal{C}' is of length 2ℓ . Therefore, we have

$$\text{dist}_{\text{TAR}}(C_0, C_r, k-1) \leq 2\ell = 2 \cdot \text{dist}_{\text{TJ}}(C_0, C_r). \quad (4)$$

We then prove that $\text{TJ}(C_0, C_r) = \text{yes}$ if $\text{TAR}(C_0, C_r, k-1) = \text{yes}$. In this case, there exists a $\text{TAR}(k-1)$ -sequence between C_0 and C_r ; let $\langle C_0, C_1, \dots, C_{\ell'} \rangle$ be a shortest one, that is, $C_{\ell'} = C_r$ and $\ell' = \text{dist}_{\text{TAR}}(C_0, C_r, k-1)$. Furthermore, by Lemma 11 we can assume that $|C_{2i-1}| = k-1$ and $|C_{2i}| = k$ for every $i \in \{1, 2, \dots, \ell'/2\}$. For every $i \in \{1, 2, \dots, \ell'/2\}$, let $C_{2i-1} = C_{2i-2} \setminus \{u_{2i-2}\}$ and $C_{2i} = C_{2i-1} \cup \{w_{2i-1}\}$. Since $\langle C_0, C_1, \dots, C_{\ell'} \rangle$ is shortest, we have $u_{2i-2} \neq w_{2i-1}$. Then, for every $i \in \{1, 2, \dots, \ell'/2\}$, we have $C_{2i-2} \leftrightarrow C_{2i}$ under TJ ; we replace each sub-sequence $\langle C_{2i-1}, C_{2i} \rangle$ with $\langle C_{2i} \rangle$, and obtain $\mathcal{C}'' = \langle C_0, C_2, C_4, \dots, C_{\ell'} \rangle$. In this way, \mathcal{C}'' is a TJ -sequence from C_0 to $C_{\ell'} = C_r$, and hence $\text{TJ}(C_0, C_r) = \text{yes}$. Furthermore, the length of \mathcal{C}'' is $\ell'/2$, and hence

$$\text{dist}_{\text{TJ}}(C_0, C_r) \leq \ell'/2 = \text{dist}_{\text{TAR}}(C_0, C_r, k-1)/2. \quad (5)$$

By Eqs. (4) and (5) we have $\text{dist}_{\text{TJ}}(C_0, C_r) = \text{dist}_{\text{TAR}}(C_0, C_r, k-1)/2$. \square

A.4 Proof of Lemma 5

Similarly as in the proof of Lemma 2, in both cases (i) and (ii), we have $C_0 \rightsquigarrow C'_0$ and $C_r \rightsquigarrow C'_r$ under $\text{TAR}(k)$. Note that $|C'_0| = |C'_r| = k+1$. Then, by Lemma 3 we have

$$\text{TJ}(C'_0, C'_r) = \text{TAR}(C'_0, C'_r, k). \quad (6)$$

We first prove that $\text{TAR}(C_0, C_r, k) = \text{yes}$ if $\text{TJ}(C'_0, C'_r) = \text{yes}$. In this case, by Eq. (6) we have $\text{TAR}(C'_0, C'_r, k) = \text{yes}$, and hence $C'_0 \rightsquigarrow C'_r$ under $\text{TAR}(k)$. Thus, $C_0 \rightsquigarrow C'_0 \rightsquigarrow C'_r \rightsquigarrow C_r$ holds under $\text{TAR}(k)$, and hence $\text{TAR}(C_0, C_r, k) = \text{yes}$.

We then prove that $\text{TJ}(C'_0, C'_r) = \text{yes}$ if $\text{TAR}(C_0, C_r, k) = \text{yes}$. In this case, since $\text{TAR}(C_0, C_r, k) = \text{yes}$, we have $C_0 \rightsquigarrow C_r$ under $\text{TAR}(k)$. Therefore, $C'_0 \rightsquigarrow C_0 \rightsquigarrow C_r \rightsquigarrow C'_r$ holds under $\text{TAR}(k)$, and hence $\text{TAR}(C'_0, C'_r, k) = \text{yes}$. By Eq. (6) we then have $\text{TJ}(C'_0, C'_r) = \text{yes}$. \square

A.5 Proof of Proposition 1

Kamiński et al. [14, Theorem 3] proved that INDEPENDENT SET RECONFIGURATION under TAR is PSPACE-complete for perfect graphs. Since the class of perfect graphs is closed under taking complements [L72], by Lemma 6 CLIQUE RECONFIGURATION under TAR is PSPACE-complete for perfect graphs. Then, Theorems 1(b) and 2(b) imply that CLIQUE RECONFIGURATION remains PSPACE-complete for perfect graphs under TS and TJ, too. \square

A.6 Proof of Proposition 1

From the definition, the class of cographs is closed under taking complements, and we note that the complement of a cograph can be computed in linear time [CPS85]. Bonsma [3] proved that INDEPENDENT SET RECONFIGURATION under TAR is solvable in linear time for cographs, and hence by Lemma 6 we can solve CLIQUE RECONFIGURATION under TAR in linear time for cographs. Then, Theorems 1(a) and 2(a) imply that CLIQUE RECONFIGURATION can be solved in linear time for cographs under TS and TJ, too. \square

B Proofs Omitted from Section 4

B.1 Proof of Theorem 3

By Theorems 1(a) and 2(a) it suffices to give an $O(w^2 n^w)$ -time algorithm for a TAR-instance; recall that the reduction from TS/TJ to TAR preserves the shortest length of reconfiguration sequences. Note that, however, the arguments for TAR below can be applied to the other rules TS and TJ, and one can obtain algorithms directly for TS and TJ rules.

Let (G, C_0, C_r, k) be any TAR-instance such that $\omega(G) \leq w$. Then, the number of cliques of size at least k in G can be bounded by $\sum_{i=k}^w \binom{n}{i} = O(n^w)$. We now construct a *reconfiguration graph* $\mathcal{R} = (\mathcal{V}, \mathcal{E})$, as follows:

- (i) each node in \mathcal{R} corresponds to a clique of G with size at least k ; and
- (ii) two nodes in \mathcal{R} are joined by an edge if and only if $C \leftrightarrow C'$ holds under $\text{TAR}(k)$ for the corresponding two cliques C and C' .

This reconfiguration graph \mathcal{R} can be constructed in time $O(w^2 n^w)$ as follows: we first enumerate all cliques in time $O(w^2 n^w)$ by checking all $O(n^w)$ vertex subsets of size at most w ; we then add edges from each clique to its $O(w)$ subsets with

one less vertex. The graph \mathcal{R} has $|\mathcal{V}| = O(n^w)$ nodes and $|\mathcal{E}| = O(wn^w)$ edges. Then, there is a $\text{TAR}(k)$ -sequence between C_0 and C_r if and only if there is a path in \mathcal{R} between the two corresponding nodes. Therefore, by the breadth-first search on \mathcal{R} which starts from the node corresponding to C_0 , we can check if \mathcal{R} has a desired path or not in time $O(|\mathcal{V}| + |\mathcal{E}|) = O(wn^w)$. Furthermore, if such a path exists, it corresponds to a shortest $\text{TAR}(k)$ -sequence between C_0 and C_r . \square

B.2 Proof of Proposition 3

We first compute a tree-decomposition \mathcal{T} with width $5t + 4$ in $O(c^t n)$ time, where c is some constant, by using the algorithm in [1]. Additionally, we can assume that the number of bags in \mathcal{T} is $O(n)$ [1]. By the definition of the tree-decomposition, every clique in G is included in at least one bag of \mathcal{T} . Since the width of \mathcal{T} is $5t + 4$, each bag in \mathcal{T} contains at most $5t + 5$ vertices of G . Thus, there are at most 2^{5t+5} cliques in each bag of \mathcal{T} , and hence we can conclude that G has $O(2^{5t+5}n)$ cliques. Then, the proposition follows, because we can construct a reconfiguration graph \mathcal{R} in time $O(t^2 2^{5t+5}n)$, similarly as in the proof of Theorem 3. \square

B.3 Proof of Lemma 7

We first prove the if-part. Suppose that there is a path $\langle M_0, M_1, \dots, M_\ell \rangle$ in $\text{MC}_k(G)$ from the node $M = M_0 \supseteq C$ to the node $M' = M_\ell \supseteq C'$. Let $C_0 = C$, and let C_j be any clique in $M_{j-1} \cap M_j$ of size k for each $j \in \{1, 2, \dots, \ell\}$; such a clique C_j exists because $|M_{j-1} \cap M_j| \geq k$. Then, $C_{j-1} \rightsquigarrow C_j$ holds under $\text{TAR}(k)$ because $C_{j-1} \cup C_j \subseteq M_{j-1}$ and hence $C_{j-1} \cup C_j$ forms a clique of G for each $j \in \{1, 2, \dots, \ell\}$. We thus have $C = C_0 \rightsquigarrow C_1 \rightsquigarrow \dots \rightsquigarrow C_\ell$ under $\text{TAR}(k)$. Since both C_ℓ and C' are contained in the same maximal clique $M_\ell = M'$, we have $C_\ell \rightsquigarrow C'$ and hence $C \rightsquigarrow C'$ holds under $\text{TAR}(k)$.

We then prove the only-if-part. Suppose that there is a $\text{TAR}(k)$ -sequence $\mathcal{C} = \langle C_0, C_1, \dots, C_{\ell'} \rangle$ such that $C_0 = C$ and $C_{\ell'} = C'$. Let $\text{MC}_k(G; \mathcal{C})$ be the subgraph of $\text{MC}_k(G)$ induced by all nodes (i.e., maximal cliques in $\mathcal{M}(G)$) that contain at least one clique in \mathcal{C} . Then, it suffices to show that $\text{MC}_k(G; \mathcal{C})$ is connected; then $\text{MC}_k(G)$ has a path from any node $M \supseteq C$ to any node $M' \supseteq C'$. Suppose for a contradiction that $\text{MC}_k(G; \mathcal{C})$ is not connected. Then, there exists an index j such that the cliques C_{j-1} and C_j are contained in different maximal cliques M_{p-1} and M_p which belong to different connected components in $\text{MC}_k(G; \mathcal{C})$. In this case, C_j must be obtained by adding a vertex u to C_{j-1} , that is, $C_j = C_{j-1} \cup \{u\}$; otherwise both C_{j-1} and C_j are contained in the same maximal clique M_{p-1} . Since \mathcal{C} is a $\text{TAR}(k)$ -sequence, we have $|C_{j-1}| \geq k$ and hence $|C_{j-1} \cap C_j| \geq k$. Then, since $C_{j-1} \subseteq M_{p-1}$ and $C_j \subseteq M_p$, we have $|M_{p-1} \cap M_p| \geq k$. Therefore, M_{p-1} and M_p must be joined by an edge in $\text{MC}_k(G)$ and hence in $\text{MC}_k(G; \mathcal{C})$. This contradicts the assumption that M_{p-1} and M_p are contained in different connected components in $\text{MC}_k(G; \mathcal{C})$. We have thus

proved that $\text{MC}_k(G; \mathcal{C})$ is connected, and hence there is a path in $\text{MC}_k(G)$ from any node $M \supseteq C$ to any node $M' \supseteq C'$. \square

C Proofs Omitted from Section 5

C.1 Proof of Lemma 8

We first prove the following lemma, which can be applied to any graph.

Lemma 12. *For two cliques C and C' in a graph G , suppose that $C \cup C'$ also forms a clique in G . Then, $\text{dist}_{\text{TAR}}(C, C', k) = |C \Delta C'|$ for every integer $k \geq \min\{|C|, |C'|\}$. Furthermore, every clique in an arbitrary shortest $\text{TAR}(k)$ -sequence from C to C' consists only of vertices in $C \cup C'$.*

Proof. We first prove that $\text{dist}_{\text{TAR}}(C, C', k) \leq |C \Delta C'|$ holds for every integer $k \geq \min\{|C|, |C'|\}$, by constructing a $\text{TAR}(k)$ -sequence between C and C' of length $|C \Delta C'|$, as follows: we first add the vertices in $C' \setminus C$ to C one by one, and obtain the clique $C \cup C'$; and we then delete the vertices in $C \setminus C'$ from $C \cup C'$ one by one, and obtain the clique C' . Since the minimum size of a clique in this sequence is $\min\{|C|, |C'|\}$, this is a $\text{TAR}(k)$ -sequence for every integer $k \geq \min\{|C|, |C'|\}$. Furthermore, the length of this $\text{TAR}(k)$ -sequence is $|C \Delta C'|$. Therefore, we have $\text{dist}_{\text{TAR}}(C, C', k) \leq |C \Delta C'|$.

We then prove that $\text{dist}_{\text{TAR}}(C, C', k) \geq |C \Delta C'|$ holds for every integer $k \geq \min\{|C|, |C'|\}$. Since $k \geq \min\{|C|, |C'|\}$, there exists at least one $\text{TAR}(k)$ -sequence between C and C' as explained above. Note that, in an arbitrary $\text{TAR}(k)$ -sequence between C and C' , every vertex in $C \Delta C'$ must be either deleted or added at least once. Therefore, we have $\text{dist}_{\text{TAR}}(C, C', k) \geq |C \Delta C'|$.

We have thus proved that $\text{dist}_{\text{TAR}}(C, C', k) = |C \Delta C'|$ holds for every integer $k \geq \min\{|C|, |C'|\}$. Consider an arbitrary shortest $\text{TAR}(k)$ -sequence \mathcal{C} from C to C' . Then, every vertex in $C \Delta C'$ must be either deleted or added by \mathcal{C} at least once. Therefore, if \mathcal{C} deletes or adds a vertex not in $C \cup C'$, then the length of \mathcal{C} is strictly greater than $|C \Delta C'|$. This contradicts the assumption that \mathcal{C} is shortest. We can thus conclude that every clique in an arbitrary shortest $\text{TAR}(k)$ -sequence from C to C' consists only of vertices in $C \cup C'$. \square

Let $G = (V, E)$ be a graph, and let $X, Y \subseteq V$. A vertex subset $S \subseteq V$ is called an (X, Y) -separator of G if any two vertices $x \in X \setminus S$ and $y \in Y \setminus S$ do not belong to the same component in $G - S$, where $G - S$ denotes the subgraph of G induced by the vertex set $V \setminus S$.

Proof of Lemma 8.

We prove the statement by induction on the length t of the unique path (M_0, M_1, \dots, M_t) in \mathcal{T} between M_0 and M_t .

First, consider the case where $t = 0$. Then, since $C_0 \subseteq M_0$ and $C_r \subseteq M_t = M_0$, both C_0 and C_r are contained in the same maximal clique M_0 . Therefore, $C_0 \cup C_r$ forms a clique, and hence by Lemma 12 every shortest $\text{TAR}(k)$ -sequence

passes through cliques consisting of vertices only in M_0 . Thus, we set $f(i) = 0$ for all $i \in \{0, 1, \dots, \ell\}$.

Next, consider the case where $t \geq 1$. We assume that $C_r \not\subseteq M_0$, because otherwise we can set $f(i) = 0$ for all $i \in \{0, 1, \dots, \ell\}$ similarly as for the case $t = 0$. Then, by the definition of a clique tree, $M_0 \cap M_1$ forms a (C_0, C_r) -separator of G (see [BP93, Lemma 4.2]).

We now claim that there exists at least one clique C_j in the shortest $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ such that $C_j \subseteq M_0 \cap M_1$. Suppose for a contradiction that $C_i \not\subseteq M_0 \cap M_1$ for all $i \in \{0, 1, \dots, \ell\}$. Let w_i be an arbitrary vertex in $C_i \setminus (M_0 \cap M_1)$ for each $i \in \{0, 1, \dots, \ell\}$. Since $\langle C_0, C_1, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence, either $C_i \subset C_{i+1}$ or $C_i \supset C_{i+1}$ holds for each $i \in \{0, 1, \dots, \ell - 1\}$ and hence $C_i \cup C_{i+1}$ forms a clique. Therefore, the vertices w_i and w_{i+1} in $C_i \cup C_{i+1}$ are either the same or adjacent. This implies that the subgraph of G induced by $\{w_i \mid 0 \leq i \leq \ell\}$ is connected, and hence it contains a path from w_0 to w_ℓ . However, since $w_0 \in C_0 \setminus (M_0 \cap M_1)$ and $w_\ell \in C_\ell \setminus (M_0 \cap M_1) = C_r \setminus (M_0 \cap M_1)$, this contradicts the assumption that $M_0 \cap M_1$ is a (C_0, C_r) -separator.

As the induction hypothesis, assume that the statement is true for the length $t - 1 \geq 0$. Let C_j be an arbitrary clique in $\langle C_0, C_1, \dots, C_\ell \rangle$ such that $C_j \subseteq M_0 \cap M_1$. Note that, since $\langle C_0, C_1, \dots, C_\ell \rangle$ is shortest, $\langle C_0, C_1, \dots, C_j \rangle$ is a shortest $\text{TAR}(k)$ -sequence from C_0 to C_j . Then, since $C_0 \cup C_j \subseteq M_0$, Lemma 12 implies that $\langle C_0, C_1, \dots, C_j \rangle$ passes through cliques consisting of vertices only in M_0 , that is,

$$C_h \subseteq M_0 \quad (7)$$

holds for each $h \in \{0, 1, \dots, j\}$. Let $C'_i = C_{j+i}$ for each $i \in \{0, 1, \dots, \ell - j\}$, and let $M'_i = M_{1+i}$ for each $i \in \{0, 1, \dots, t - 1\}$. Note that $\langle C'_0, C'_1, \dots, C'_{\ell-j} \rangle$ is a shortest $\text{TAR}(k)$ -sequence from $C'_0 = C_j$ to $C'_{\ell-j} = C_\ell = C_r$. Furthermore, $C'_0 = C_j \subseteq M_1 = M'_0$, $C'_{\ell-j} = C_r \subseteq M_t = M'_{t-1}$ and $(M'_0, M'_1, \dots, M'_{t-1})$ is a path in \mathcal{T} of length $t - 1$. Therefore, by the induction hypothesis, there is a monotonically increasing function $f': \{0, 1, \dots, \ell - j\} \rightarrow \{0, 1, \dots, t - 1\}$ such that

$$C'_i \subseteq M'_{f'(i)} \quad (8)$$

for all $i \in \{0, 1, \dots, \ell - j\}$. Now we construct a mapping $f: \{0, 1, \dots, \ell\} \rightarrow \{0, 1, \dots, t\}$, as follows:

$$f(i) = \begin{cases} 0 & \text{if } i < j, \\ f'(i - j) + 1 & \text{otherwise.} \end{cases}$$

Since f' is a monotonically increasing function, f is too. Furthermore, by Eqs. (7) and (8) we have $C_i \subseteq M_{f(i)}$ for all $i \in \{0, 1, \dots, \ell\}$. Thus, f satisfies the desired property. \square

C.2 Proof of Lemma 9

Before giving our linear-time reduction, we give the following lemma.

Lemma 13. *Suppose that $\langle C_0, C_1, \dots, C_\ell \rangle$ is a shortest $\text{TAR}(k)$ -sequence in a chordal graph G . Let p and q be two indices in $\{0, 1, \dots, \ell\}$ such that $p < q$. If there is a vertex v in $C_p \cap C_q$, then $v \in C_i$ holds for all $i \in \{p, p+1, \dots, q\}$.*

Proof. Suppose for a contradiction that the statement does not hold. We may assume without loss of generality that $v \notin C_i$ for every $i \in \{p+1, p+2, \dots, q-1\}$ by setting p as large as possible and q as small as possible. Then, observe that $C_{p+1} \cup \{v\} = C_p$ and $C_{q-1} \cup \{v\} = C_q$.

Let \mathcal{T} be a clique tree of G . Let (M_0, M_1, \dots, M_t) be the path in \mathcal{T} from M_0 to M_t for any pair of maximal cliques $M_0 \supseteq C_0$ and $M_t \supseteq C_\ell$. By Lemma 8 there is a monotonically increasing function $f: \{0, 1, \dots, \ell\} \rightarrow \{0, 1, \dots, t\}$ such that $C_i \subseteq M_{f(i)}$ for each $i \in \{0, 1, \dots, \ell\}$. Then, $f(p) \leq f(i) \leq f(q)$ for each $i \in \{p+1, p+2, \dots, q-1\}$. Recall that, by the definition of a clique tree, the subgraph of \mathcal{T} induced by $\mathcal{M}(G; v)$ is connected. Since $v \in C_p \cap C_q \subseteq M_{f(p)} \cap M_{f(q)}$, we can conclude that the vertex v is contained in all maximal cliques $M_{f(p)}, M_{f(p+1)}, \dots, M_{f(q)}$.

Therefore, for each $i \in \{p, p+1, \dots, q\}$, both $C_i \subseteq M_{f(i)}$ and $v \in M_{f(i)}$ hold, and hence $C_i \cup \{v\}$ forms a clique which is contained in $M_{f(i)}$. Furthermore, $C_i \cup \{v\} \leftrightarrow C_{i+1} \cup \{v\}$ under $\text{TAR}(k)$ for each $i \in \{p, p+1, \dots, q-1\}$, because $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$. Recall that $C_{p+1} \cup \{v\} = C_p$ and $C_{q-1} \cup \{v\} = C_q$, and hence we replace the sub-sequence $\langle C_p, C_{p+1}, \dots, C_q \rangle$ of length $q - p + 1$ with the following sequence of length $q - p - 1$:

$$\langle C_{p+1} \cup \{v\}, C_{p+2} \cup \{v\}, \dots, C_{q-1} \cup \{v\} \rangle.$$

However, this contradicts the assumption that $\langle C_0, C_1, \dots, C_\ell \rangle$ is shortest. \square

Proof of Lemma 9.

We first add two dummy vertices d_0 and d_r to a given chordal graph G . We then join d_0 with all vertices in C_0 by adding new edges to G ; similarly, we join d_r with all vertices in C_r . Let G' be the resulting graph. Then, G' is also a chordal graph, because the dummy vertices cannot create any new induced cycle of length more than three. Note that each of $C_0 \cup \{d_0\}$ and $C_r \cup \{d_r\}$ forms a maximal clique in G' . Furthermore, in the set $\mathcal{M}(G')$ of all maximal cliques in G , the only maximal cliques $C_0 \cup \{d_0\}$ and $C_r \cup \{d_r\}$ contain d_0 and d_r , respectively.

We now construct a clique tree \mathcal{T}' of G' in linear time [18, §15.1]. Then, \mathcal{T}' contains two nodes $M_0 = C_0 \cup \{d_0\}$ and $M_t = C_r \cup \{d_r\}$. Therefore, we can find the path (M_0, M_1, \dots, M_t) in \mathcal{T}' in linear time. Let H'' be the subgraph of G induced by the maximal cliques M_0, M_1, \dots, M_t . Then, H'' is an interval graph. Furthermore, since $M_0 = C_0 \cup \{d_0\}$ and $M_t = C_r \cup \{d_r\}$, Lemma 8 implies that

$$\text{dist}_{\text{TAR}}(H'', C_0, C_r, k) = \text{dist}_{\text{TAR}}(G', C_0, C_r, k). \quad (9)$$

Let H be the graph obtained from H'' by removing the dummy vertices d_0 and d_r . Since H'' is an interval graph, H is also an interval graph. In this way, H can be constructed in linear time.

Now we claim that

$$\text{dist}_{\text{TAR}}(G', C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k) \quad (10)$$

and

$$\text{dist}_{\text{TAR}}(H, C_0, C_r, k) = \text{dist}_{\text{TAR}}(H'', C_0, C_r, k). \quad (11)$$

Then, by Eqs. (9)–(11) we have $\text{dist}_{\text{TAR}}(H, C_0, C_r, k) = \text{dist}_{\text{TAR}}(G, C_0, C_r, k)$, as required. Note that $V(G) \Delta V(G') = V(H) \Delta V(H'') = \{d_0, d_r\}$. Thus, to prove Eqs. (10) and (11), it suffices to show that there is a shortest $\text{TAR}(k)$ -sequence in G' (or in H'') from C_0 to C_r which does not pass through any clique containing d_0 or d_r .

Let $\langle C_0, C_1, \dots, C_\ell \rangle$ be a shortest $\text{TAR}(k)$ -sequence in G' (or in H'') from C_0 to $C_\ell = C_r$. Suppose for a contradiction that $d_0 \in C_i$ holds for some $i \in \{1, 2, \dots, \ell - 1\}$. (The proof for d_r is the same.) Since $d_0 \notin C_0 \cup C_\ell$, Lemma 13 implies that there exists a pair of indices l and r in $\{1, 2, \dots, \ell - 1\}$ such that $l \leq r$ and $d_0 \in C_i$ holds for all $i \in \{l, l + 1, \dots, r\}$. Recall that $C_0 \cup \{d_0\}$ is a maximal clique in G' (or in H''), and that no other maximal clique in G' (or in H'') contains d_0 . This implies that $C_i \subseteq C_0 \cup \{d_0\}$ for each $i \in \{l, l + 1, \dots, r\}$. Since $C_{l-1} = C_l \setminus \{d_0\}$ and $C_{r+1} = C_r \setminus \{d_0\}$, it follows that $C_{l-1} \cup C_{r+1} \subseteq C_0$ and hence $C_{l-1} \cup C_{r+1}$ forms a clique. Now, by Lemma 12 every shortest $\text{TAR}(k)$ -sequence from C_{l-1} to C_{r+1} passes through cliques consisting of vertices only in $C_{l-1} \cup C_{r+1} \subseteq C_0$. Since $d_0 \notin C_0$, this contradicts the assumption that $\langle C_{l-1}, C_l, \dots, C_{r+1} \rangle$ is shortest. \square

C.3 Correctness of the algorithm for interval graphs

In this subsection, we prove the correctness of the greedy algorithm in Section 5.2 and estimate its running time. For a vertex v in a graph G , let $N(v) = \{w \in V(G) \mid vw \in E(G)\}$ and let $N[v] = N(v) \cup \{v\}$. We denote by $\deg(v)$ the degree of v , that is, $\deg(v) = |N(v)|$.

We first prove the correctness of Step (1) of the algorithm: if $C_0 \not\subseteq C_r$ and $|C_0| \geq k + 1$, then remove a vertex u with the minimum r -value in $C_0 \setminus C_r$ from C_0 . The following lemma ensures that this operation preserves the shortest length of reconfiguration sequences.

Lemma 14. *Suppose that $C_0 \not\subseteq C_r$ and $|C_0| \geq k + 1$. Let u be any vertex with the minimum r -value in $C_0 \setminus C_r$. Then,*

$$\text{dist}_{\text{TAR}}(C_0, C_r, k) = \text{dist}_{\text{TAR}}(C_0 \setminus \{u\}, C_r, k) + 1.$$

Proof. First, observe that $r_u = 0$ since $C_0 \not\subseteq M_1$. Thus, $N[u] = M_0 \subseteq N[v]$ holds for every vertex $v \in M_0$. Consider any clique C in H such that $C_0 \leftrightarrow C$ under $\text{TAR}(k)$. Then, either (i) $C = C_0 \setminus \{v\}$ for some vertex $v \in C_0$, or (ii) $C = C_0 \cup \{w\}$ for some vertex $w \in M_0 \setminus C_0$; recall that $C_0 \subseteq M_0$ and $C_0 \not\subseteq M_1$. Therefore, it suffices to verify the following two inequalities:

$$\text{dist}_{\text{TAR}}(C_0 \setminus \{u\}, C_r, k) \leq \text{dist}_{\text{TAR}}(C_0 \setminus \{v\}, C_r, k) \quad (12)$$

for any vertex $v \in C_0$; and

$$\text{dist}_{\text{TAR}}(C_0 \setminus \{u\}, C_r, k) \leq \text{dist}_{\text{TAR}}(C_0 \cup \{w\}, C_r, k) \quad (13)$$

for any vertex $w \in M_0 \setminus C_0$.

We first prove Eq. (12). Let v be any vertex in $C_0 \setminus \{u\}$, and let $\langle C_1, C_2, \dots, C_\ell \rangle$ be a shortest $\text{TAR}(k)$ -sequence from $C_1 = C_0 \setminus \{v\}$ to $C_\ell = C_r$. By Lemma 13 we have $v \notin C_i$ for all $i \in \{1, 2, \dots, \ell\}$. On the other hand, since $u \in C_1 \setminus C_\ell$, there exists an index $j \geq 1$ such that $C_{j+1} = C_j \setminus \{u\}$; Lemma 13 implies that $u \in C_i$ if and only if $i \in \{1, 2, \dots, j\}$. Then, $C'_i = (C_i \setminus \{u\}) \cup \{v\}$ forms a clique for each $i \in \{1, 2, \dots, j\}$, because $N[u] \subseteq N[v]$ for the vertex $v \in C_0 \setminus \{u\} \subset M_0$. For each $i \in \{1, 2, \dots, j\}$, we replace the clique C_i in $\langle C_1, C_2, \dots, C_\ell \rangle$ with the clique $C'_i = (C_i \setminus \{u\}) \cup \{v\}$, and obtain the following sequence \mathcal{C}' of cliques:

$$\mathcal{C}' = \langle C'_1, C'_2, \dots, C'_j, C_{j+1}, C_{j+2}, \dots, C_\ell \rangle.$$

Since $\langle C_1, C_2, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence, we have $|C'_i| = |C_i| \geq k$. Furthermore, $C'_i \leftrightarrow C'_{i+1}$ under $\text{TAR}(k)$ for all $i \in \{1, 2, \dots, j-1\}$, since $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$. Finally, since $C_{j+1} = C_j \setminus \{u\}$, we have $C'_j \setminus \{v\} = C_{j+1}$ and hence $C'_j \leftrightarrow C_{j+1}$ under $\text{TAR}(k)$. Therefore, \mathcal{C}' is a $\text{TAR}(k)$ -sequence from $C'_1 = C_0 \setminus \{u\}$ to $C_\ell = C_r$, which has the same length ℓ as the shortest $\text{TAR}(k)$ -sequence $\langle C_1, C_2, \dots, C_\ell \rangle$ from $C_1 = C_0 \setminus \{v\}$ to $C_\ell = C_r$. We have thus verified Eq. (12).

We then prove Eq. (13). Let w be any vertex in $M_0 \setminus C_0$, and let $\langle C_1, C_2, \dots, C_\ell \rangle$ be a shortest $\text{TAR}(k)$ -sequence from $C_1 = C_0 \cup \{w\}$ to $C_\ell = C_r$. Let $j \in \{1, 2, \dots, \ell-1\}$ be the index such that $u \in C_i$ if and only if $i \in \{1, 2, \dots, j\}$. Since $r_u = 0$, all cliques C_1, C_2, \dots, C_j are contained in M_0 . Furthermore, since $C_{j+1} = C_j \setminus \{u\}$, we have $C_{j+1} \subseteq M_0$ and hence $C_1 \cup C_{j+1} (\subseteq M_0)$ forms a clique. Then, Lemma 12 implies that $\text{dist}_{\text{TAR}}(C_1, C_{j+1}, k) = |C_1 \Delta C_{j+1}|$. Note that, since the sub-sequence $\langle C_1, C_2, \dots, C_{j+1} \rangle$ is shortest, we have $\text{dist}_{\text{TAR}}(C_1, C_{j+1}, k) = |C_1 \Delta C_{j+1}| = j$. On the other hand, consider the clique $C'_1 = C_0 \setminus \{u\}$; note that, since $|C_0| \geq k+1$, we have $|C'_1| \geq k$. Since $C'_1, C_{j+1} \subseteq M_0$, the set $C'_1 \cup C_{j+1}$ forms a clique. Then, Lemma 12 implies that $\text{dist}_{\text{TAR}}(C'_1, C_{j+1}, k) = |C'_1 \Delta C_{j+1}|$. We now prove that

$$\text{dist}_{\text{TAR}}(C'_1, C_{j+1}, k) \leq \text{dist}_{\text{TAR}}(C_1, C_{j+1}, k) = j. \quad (14)$$

Indeed, we show that $|C'_1 \Delta C_{j+1}| \leq |C_1 \Delta C_{j+1}|$, as follows. Since $C'_1 = C_1 \setminus \{u, w\}$, $u, w \in C_1$ and $u \notin C_{j+1}$, we have

$$\begin{aligned} |C'_1 \Delta C_{j+1}| &= |C'_1 \cup C_{j+1}| - |C'_1 \cap C_{j+1}| \\ &= |(C_1 \setminus \{u, w\}) \cup C_{j+1}| - |(C_1 \setminus \{u, w\}) \cap C_{j+1}| \\ &= \begin{cases} (|C_1 \cup C_{j+1}| - |\{u\}|) - (|C_1 \cap C_{j+1}| - |\{w\}|) & \text{if } w \in C_{j+1}, \\ (|C_1 \cup C_{j+1}| - |\{u, w\}|) - |C_1 \cap C_{j+1}| & \text{if } w \notin C_{j+1} \end{cases} \\ &\leq |C_1 \cup C_{j+1}| - |C_1 \cap C_{j+1}| \\ &= |C_1 \Delta C_{j+1}|. \end{aligned}$$

Let $\langle C'_1, C'_2, \dots, C'_j, C_{j+1} \rangle$ be a shortest $\text{TAR}(k)$ -sequence from C'_1 to C_{j+1} . Then, by Eq. (14) the length of $\langle C'_1, C'_2, \dots, C'_j, C_{j+1} \rangle$ is at most j . We replace the sub-sequence $\langle C_1, C_2, \dots, C_j, C_{j+1} \rangle$ of length j with the $\text{TAR}(k)$ -sequence $\langle C'_1, C'_2, \dots, C'_j, C_{j+1} \rangle$. Then, $\langle C'_1, C'_2, \dots, C'_j, C_{j+1}, C_{j+2}, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence from $C'_1 = C_0 \setminus \{u\}$ to $C_\ell = C_r$, whose length is at most $\ell - 1 = \text{dist}_{\text{TAR}}(C_0 \cup \{u\}, C_r, k)$. We have thus verified Eq. (13). \square

We then prove the correctness of Step (2) of the algorithm: if no vertex can be deleted from C_0 according to Lemma 14, then add a vertex u chosen by the following lemma, with preserving the shortest length of reconfiguration sequences.

Lemma 15. *Assume that $C_0 \subseteq C_r$ or $|C_0| = k$. Let u be any vertex in $(C_r \setminus C_0) \cap M_0$ if exists; otherwise, let u be any vertex with the maximum r -value in $M_0 \setminus C_0$. Then,*

$$\text{dist}_{\text{TAR}}(C_0, C_r, k) = \text{dist}_{\text{TAR}}(C_0 \cup \{u\}, C_r, k) + 1.$$

Proof. Note that, if $|C_0| = k$, then no vertex can be deleted from C_0 due to the size constraint k . On the other hand, if $C_0 \subseteq C_r$, then by Lemma 13 no shortest $\text{TAR}(k)$ -sequence from C_0 to C_r deletes any vertex v in C_0 , because $v \in C_0 \cap C_r$. Therefore, in any shortest $\text{TAR}(k)$ -sequence $\langle C_0, C_1, \dots, C_\ell \rangle$ from C_0 to $C_\ell = C_r$, the clique C_1 must be obtained from C_0 by adding a vertex $v \in V(G) \setminus C_0$. Furthermore, since $C_0 \subseteq M_0$, $C_0 \not\subseteq M_1$ and $C_1 = C_0 \cup \{v\}$ is a clique, the added vertex v must be in $M_0 \setminus C_0$. Thus, to prove the lemma, it suffices to show that

$$\text{dist}_{\text{TAR}}(C_0 \cup \{u\}, C_r, k) \leq \text{dist}_{\text{TAR}}(C_0 \cup \{v\}, C_r, k) \quad (15)$$

for any vertex $v \in M_0 \setminus C_0$.

Let v be any vertex in $M_0 \setminus C_0$, and let $\langle C_1, C_2, \dots, C_\ell \rangle$ be a shortest $\text{TAR}(k)$ -sequence from $C_1 = C_0 \cup \{v\}$ to $C_\ell = C_r$. For each $i \in \{1, 2, \dots, \ell\}$, let

$$C'_i = \begin{cases} (C_i \setminus \{v\}) \cup \{u\} & \text{if } v \in C_i \text{ and } u \notin C_i, \\ C_i & \text{otherwise.} \end{cases} \quad (16)$$

We will prove below that $\langle C'_1, C'_2, \dots, C'_\ell \rangle$ is a $\text{TAR}(k)$ -sequence from $C_0 \cup \{u\}$ to C_r . Then, since $\langle C'_1, C'_2, \dots, C'_\ell \rangle$ is of length $\ell = \text{dist}_{\text{TAR}}(C_0 \cup \{v\}, C_r, k)$, Eq. (15) follows.

We first claim that $C'_1 = C_0 \cup \{u\}$ and $C'_\ell = C_\ell$. Since $v \in C_0 \cup \{v\} = C_1$ and $u \notin C_0 \cup \{v\} = C_1$, we have $C'_1 = (C_1 \setminus \{v\}) \cup \{u\} = C_0 \cup \{u\}$. On the other hand, if u is chosen from $(C_r \setminus C_0) \cap M_0$, then $u \in C_r = C_\ell$ and hence $C'_\ell = C_\ell$. Otherwise, $(C_r \setminus C_0) \cap M_0 = (M_0 \setminus C_0) \cap C_r = \emptyset$ holds, and hence $v \in M_0 \setminus C_0$ is not contained in $C_r = C_\ell$; we then have $C'_\ell = C_\ell$.

We then prove that C'_i forms a clique of size at least k for each $i \in \{1, 2, \dots, \ell\}$, and prove that $C'_i \leftrightarrow C'_{i+1}$ under $\text{TAR}(k)$ for each $i \in \{1, 2, \dots, \ell - 1\}$. Since $\langle C_1, C_2, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence, by Eq. (16) we have $|C'_i| = |C_i| \geq k$.

Therefore, it suffices to show that $C'_i \cup C'_{i+1}$ forms a clique such that $|C'_i \Delta C'_{i+1}| = 1$ for each $i \in \{1, 2, \dots, \ell - 1\}$. This claim trivially holds for the case where both $C'_i = C_i$ and $C'_{i+1} = C_{i+1}$ hold, because $\langle C_1, C_2, \dots, C_\ell \rangle$ is a $\text{TAR}(k)$ -sequence. By symmetry, we thus assume that $C'_i = (C_i \setminus \{v\}) \cup \{u\}$, that is, both $v \in C_i$ and $u \notin C_i$ hold. Then, there are the following three cases to consider; note that, since both $v \in C_i$ and $u \notin C_i$ hold and $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$, we do not need to consider the case where both $v \notin C_{i+1}$ and $u \in C_{i+1}$ hold.

Case (i) $v \in C_{i+1}$ and $u \notin C_{i+1}$.

In this case, we have $C'_{i+1} = (C_{i+1} \setminus \{v\}) \cup \{u\}$. Since $C'_i = (C_i \setminus \{v\}) \cup \{u\}$ and $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$, we have $|C'_i \Delta C'_{i+1}| = |C_i \Delta C_{i+1}| = 1$. Notice that $l_v = l_u = 0$ and $r_v \leq r_u$, because $r_u = t$ or u has the maximum r -value in $M_0 \setminus C_0$. Therefore, $N[v] \subseteq N[u]$ holds. Then, since $C_i \cup C_{i+1}$ is a clique, $C'_i \cup C'_{i+1} = ((C_i \cup C_{i+1}) \setminus \{v\}) \cup \{u\}$ forms a clique.

Case (ii) $v, u \in C_{i+1}$.

In this case, we have $C'_{i+1} = C_{i+1}$. Recall that both $v \in C_i$ and $u \notin C_i$ hold. Then, since $v, u \in C_{i+1}$ and $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$, we have $C_i \cup \{u\} = C_{i+1} = C'_{i+1}$. Since $C'_i = (C_i \cup \{u\}) \setminus \{v\}$, we thus have $C'_i = C'_{i+1} \setminus \{v\}$ and hence $|C'_i \Delta C'_{i+1}| = |\{v\}| = 1$. Furthermore, since $C'_{i+1} = C_{i+1}$ and C_{i+1} is a clique, $C'_i \cup C'_{i+1} = C'_{i+1}$ forms a clique.

Case (iii) $v, u \notin C_{i+1}$.

In this case, we have $C'_{i+1} = C_{i+1}$. Recall again that both $v \in C_i$ and $u \notin C_i$ hold. Then, since $v, u \notin C_{i+1}$ and $C_i \leftrightarrow C_{i+1}$ under $\text{TAR}(k)$, we have $C_i \setminus \{v\} = C_{i+1} = C'_{i+1}$. Since $C'_i = (C_i \setminus \{v\}) \cup \{u\}$, we thus have $C'_i = C'_{i+1} \cup \{u\}$ and hence $|C'_i \Delta C'_{i+1}| = |\{u\}| = 1$. Then, $C'_i \cup C'_{i+1} = C'_i = (C_i \setminus \{v\}) \cup \{u\}$. Since $N[v] \subseteq N[u]$ holds and C_i is a clique, $C'_i \cup C'_{i+1} = (C_i \setminus \{v\}) \cup \{u\}$ forms a clique.

In this way, we have proved that $\langle C'_1, C'_2, \dots, C'_\ell \rangle$ is a $\text{TAR}(k)$ -sequence from $C_0 \cup \{u\}$ to C_r , and hence Eq. (15) holds as we have mentioned above. \square

The correctness of the greedy algorithm in Section 5.2 follows from Lemmas 14 and 15. Therefore, to complete the proof of Theorem 5, we now show that the algorithm runs in linear time.

Estimation of the running time.

Lemma 13 implies that each vertex is removed at most once and added at most once in any shortest $\text{TAR}(k)$ -sequence. Therefore, it suffices to show that each removal and addition of a vertex u can be done in time $O(\deg(u))$, because $\sum_{u \in V(G)} \deg(u) = 2|E(H)|$.

We first estimate the running time for Step (1) of the algorithm. We first check whether both $C_0 \not\subseteq C_r$ and $|C_0| \geq k + 1$ hold or not. These conditions can be checked in constant time by maintaining $|C_0|$ and $|C_0 \cap C_r|$. We then find a vertex u with the minimum r -value in $C_0 \setminus C_r$; this can be done in time $O(|C_0|)$. After the removal of u , the clique $C_0 := C_0 \setminus \{u\}$ may be included by some of M_1, M_2, \dots, M_t ; in such a case, we need to shift the indices of M_i

so that $C_0 \subseteq M_0$ and $C_0 \not\subseteq M_1$ hold. To do so, we compute the shift-value $i_0 = \min\{r_u \mid u \in C_0\}$, and set $M_i := M_{i-i_0}$ for each $i \in \{1, 2, \dots, t\}$ and $r_w := r_w - i_0$ for each vertex $w \in V(H)$. However, since we just have to compute and store only the shift-value i_0 in the actual process, this post-process can be done also in time $O(|C_0|)$. Since $C_0 \subseteq N[u]$, we have $|C_0| \leq \deg(u)+1$. Therefore, Step (1) can be executed in time $O(\deg(u))$.

We then estimate the running time for Step (2) of the algorithm. We find a vertex u which either is in $(C_r \setminus C_0) \cap M_0$ or has the maximum r -value in $M_0 \setminus C_0$. In either case, such a vertex u can be found in time $O(|M_0|)$. Since $M_0 \subseteq N[u]$, the addition of u can be done in time $O(\deg(u))$.

References

- [BP93] Blair, J.R.S., Peyton, B.: An introduction to chordal graphs and clique trees. *Graph Theory and Sparse Matrix Computation* 56, pp. 1–29 (1993)
- [CPS85] Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for cographs. *SIAM J. Computing* 14, pp. 926–934 (1985)
- [L72] Lovász, L.: Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics* 2, pp. 253–267 (1972)